Abstract

In this paper, an implicit finite difference method based on the Crank–Nicolson method is proposed for the numerical solution of the one-dimensional Burger–Fisher equation. The Crank–Nicolson scheme provides a system of nonlinear difference equations, which is solved by an integration of the Jacobian-Free-Newton-Krylov (JFNK) and GMRES methods. Various numerical examples are given to demonstrate the efficiency of the proposed scheme. Comparison of the computed solutions with the analytical ones demonstrates the accuracy of this proposed method.

Keywords: Crank–Nicolson scheme; burger–fisher equation; jacobian-free-newton-krylov method; GMRES method.

1 Introduction

Being a combination of convection, diffusion, and reaction mechanisms, the Burger–Fisher equation is highly nonlinear. This equation has many applications in many scientific fields such as gas dynamics, number theory, elasticity, heat conduction, etc. [1]. Recently, several methods have been proposed to solve it. Here, we briefly discuss the methods of some researchers.


A generalized form of the one-dimensional Burger–Fisher equation is as follows.

\[
\begin{align*}
\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2} + \alpha(t) U^\delta \frac{\partial U}{\partial x} &= \beta(t) U(1 - U^\delta), x \in \Omega, 0 \leq t \leq T, \\
\end{align*}
\]  

(1)

where \(\Omega = \{ x | a \leq x \leq b \text{ and } a, b \in \mathbb{R}\}\), \(\alpha(t)\) and \(\beta(t)\) are continuous functions, and \(U(x, t)\) is an unknown continuous function. The initial and boundary conditions of this problem are as follows.

\[
\begin{align*}
U(x, 0) &= \Phi(x), \\
U(a, t) &= \Psi_1(x, t), U(b, t) = \Psi_2(x, t), t > 0.
\end{align*}
\]

(2)

The remainder of this paper is structured as follows. In Section 2, the numerical method is explained. Some numerical experiments are reported in Section 3. Section 4 is dedicated to the conclusion.

2. The Method of Solution

In this section, the discretization of equation (1) is explained, and the Jacobian-Free-Newton-Krylov (JFNK) method is explained to solve the obtained system of nonlinear equations.

2.1 The Discretization Method

Consider \(\Delta x\) as a grid size in the space such that \(\{x_i | x_i = i \Delta x, i = 0, 1, \ldots, l\}\) covers \(\Omega\). Consider a positive integer \(N\). The grid size in time \((\Delta t)\) for the finite difference scheme is \(\frac{T}{N}\). Consider \(U^n_i\) as the value of \(U(x_i, t_n)\). The Crank–Nicolson approximation of (1) can be written as follows.

\[
\begin{align*}
\frac{U^n_i - U^{n-1}}{\Delta t} - \frac{1}{2} \left[ \frac{U^n_{i+1} - 2U^n_i + U^n_{i-1}}{(\Delta x)^2} + \frac{U^{n-1}_{i+1} - 2U^{n-1}_i + U^{n-1}_{i-1}}{(\Delta x)^2} \right] + \frac{1}{2} \left[ \alpha(t^n)(U^{\delta n}_{i+1} - U^n_i) \frac{(\Delta x)^2}{(\Delta x)} + \alpha(t^{n-1})(U^{\delta n-1}_{i+1} - U^{n-1}_i) \frac{(\Delta x)^2}{(\Delta x)} \right] = \frac{1}{2} \left[ \beta(t^n)U^n_i(1 - (U^\delta^n)^n) + \beta(t^{n-1})U^{n-1}_i(1 - (U^\delta^{n-1})^{n-1}) \right],
\end{align*}
\]

(4)

To solve the system (4) of nonlinear equations, the JFNK method is used together with the GMRES method. The main advantage of the JFNK method [24] is that using it, no cost is incurred for the creation and storage of the Jacobian matrix.
Algorithm 1 gives one iteration of this integrated method [29]

<table>
<thead>
<tr>
<th>Algorithm 1[29]: $u^{r+1} := JFNK(F, u^r)$</th>
</tr>
</thead>
</table>
| /* This Algorithm solves $A\delta u^r = b$ by the GMRES method. where $A = J^r$, $b = -F(u^r)$, $u^r$ is the approximate solution for $F(u) = 0$. Then it updates $u^r */$
| /* $e_1$ is the unit vector: $e_1 = [1 \ 0 \ \ldots \ 0]^T */$
| /* $H$ is Hessenberg matrix: $H = \{h_{ij}\} */$
| /* $V$ is unitary matrix: $V = [v_1, v_2, \ldots ] */$
| 1. Choose $\varepsilon_G$ /* A tolerance to stop the GMRES method */
| 2. Choose $\delta u^r$ /* An initial approximate solution */
| (In this experience $\delta u^r = 0$ )
| 3. $\varepsilon := 10^{-8}$
| 4. $r_0 := -F(u^r) - \frac{f(u^r + \delta u^r) - F(u^r)}{\varepsilon}$
| 5. $y := \|r_0\|_2$
| 6. $v_1 = \frac{r_0}{y}$
| 7. For $j = 1, 2, \ldots$, until convergence, Do
| 8. $\delta u^r := u^r + \delta u^r$. End. |

2.2 The JFNK Method

The JFNK method integrates the Newton method to solve a system of nonlinear equations and a Krylov subspace method to solve the system of linear equations resulted from the Newton method. In this method, the creation and storage of the Jacobian matrix are not required. The integration of the JFNK with GMRES methods is described in [25]. Assume a system of nonlinear equations as follows

$F(u) = 0,$

where $u$ is the unknown vector. In each iteration $r$ of the Newton method, the system of linear equations

$J^r \delta u^r = -F(u^r), \quad (5)$

where $J^r$ is the Jacobian matrix in iteration $r$, is solved by the GMRES method. Then, the approximate solution is updated as

$u^{r+1} = u^r + \delta u^r, r = 0, 1, 2, \ldots$, 

where $u^0$ is an initial approximate solution for the Newton method. The Newton method is stopped by the following criterion
\[ \| F(u^{k+1}) \| \leq \varepsilon_{\text{Newton}}, \]

where \( \varepsilon_{\text{Newton}} \) is a tolerance for stopping the Newton method. In the JFNK method, the product \( J^*x_j \) is approximated as

\[ J^*x_j \approx \frac{F(u^{k+1}+x_j)-F(u^k)}{\varepsilon}. \]

The value of \( \varepsilon \) can be determined by various approaches. See [26], [27], and [28]. In this work, as in [25], the following value is used.

\[ \varepsilon = 10^{-8} \frac{\| u^k \|_2}{\| x_j \|_2}. \]

3 Numerical Experiments

In this section, numerical solutions of problem (1) with initial and boundary conditions (2)-(3) are computed using the approximation scheme (4). Then, these are compared with the analytical solutions. The accuracy of this proposed, numerical method is measured in terms of the relative error

\[ L_r(t) = \max_{t \in [0,1]} \left| \frac{\hat{U}(x, t) - U(x, t)}{U(x, t)} \right|, \]

where \( \hat{U} \) and \( U \) are the numerical approximation and the exact solution, respectively.

Example 1. We know from [11] that, if \( \alpha, \beta, \delta \in \mathbb{R} \) then the exact solution of equation (1) is

\[ U(x, t) = \frac{1}{2} + \frac{1}{2} \tanh \left( \theta_1 (x - \theta_2 t) \right), \] (6)

where \( \theta_1 = \frac{-\alpha \delta}{2(1+\delta)} \) and \( \theta_2 = \frac{\alpha}{1+\delta} + \frac{\beta(1+\delta)}{\alpha} \). Consider equation (1) for \( x \in [0,1] \), with initial and boundary conditions as in (6).

In Table 1, the maximum relative errors at \( t = 0.5 \) and \( t = 1 \) with \( \Delta x = \Delta t = 0.1 \) for \( \delta = 1 \) and various values of \( \alpha \) and \( \beta \) are presented. In Fig. 1, the function \( U \) introduced in relation (6) is plotted with \( \beta = 8, \alpha = 10^{-4}, \) and \( \delta = 1 \).
Example 2. We know from [30] that the exact solution of equation (1) is

\[ U(x, t) = \frac{\exp(\delta x - k^2 t) + 1.5 - \exp(-\delta x + k^2 t)}{\exp(\delta x - k^2 t) + 2}, \]  

(7)

where \( \delta = 1 \), \( k \) and \( \beta \) are arbitrary, and \( \alpha = \frac{\beta}{k} \). Consider equation (1) for \( x \in [0, 5] \), with initial and boundary conditions as in (7). Table 2 shows the maximum relative errors at \( t = 0.5 \) and \( t = 1 \) with \( \beta = 100 + \cos(t) \) and various values of \( k, \Delta t, \) and \( \Delta x \). In Figs 2 and 3, the function \( U \) introduced in relation (7) is plotted with \( k = 3, 4 \), respectively.

Table 2. The maximum relative errors at \( t = 0.5 \) and \( t = 1 \) with \( \beta = 100 + \cos(t) \), various values of \( k, \Delta t, \) and \( \Delta x, \) and \( \alpha = \frac{\beta}{k} \) in Example 2

<table>
<thead>
<tr>
<th>( t = 0.5 )</th>
<th>( K = 3 )</th>
<th>( K = 4 )</th>
<th>( t = 1 )</th>
<th>( K = 3 )</th>
<th>( K = 4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta t = \Delta x = 0.01 )</td>
<td>( L_{r_e} = 3.2864e-5 )</td>
<td>( L_{r_e} = 1.4554e-4 )</td>
<td>( L_{r_e} = 8.9482e-05 )</td>
<td>( L_{r_e} = 3.0119e-4 )</td>
<td>( L_{r_e} = 1.3769e-4 )</td>
</tr>
<tr>
<td>( \Delta t = \Delta x = 0.004 )</td>
<td>( L_{r_e} = 5.5480e-6 )</td>
<td>( L_{r_e} = 1.8431e-5 )</td>
<td>( L_{r_e} = 1.0789e-5 )</td>
<td>( L_{r_e} = 4.1730e-5 )</td>
<td>( L_{r_e} = 1.0138e-5 )</td>
</tr>
<tr>
<td>( \Delta t = \Delta x = 0.002 )</td>
<td>( L_{r_e} = 2.0943e-6 )</td>
<td>( L_{r_e} = 4.5961e-6 )</td>
<td>( L_{r_e} = 2.2481e-6 )</td>
<td>( L_{r_e} = 1.0138e-5 )</td>
<td>( L_{r_e} = 1.0138e-5 )</td>
</tr>
</tbody>
</table>

Table 3. The maximum relative errors at \( t = 0.5 \) and \( t = 1 \) with \( \beta = 6t^5 \) and various values of \( k, \Delta t, \) and \( \Delta x, \) and \( \alpha = \frac{\beta}{k} \) in Example 3

<table>
<thead>
<tr>
<th>( t = 0.5 )</th>
<th>( K = 2 )</th>
<th>( K = 3 )</th>
<th>( K = 4 )</th>
<th>( t = 1 )</th>
<th>( K = 3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta t = \Delta x = 0.01 )</td>
<td>( L_{r_e} = 9.2685e-5 )</td>
<td>( L_{r_e} = 6.2437e-4 )</td>
<td>( L_{r_e} = 1.8225e-4 )</td>
<td>( L_{r_e} = 8.3802e-4 )</td>
<td>( L_{r_e} = 1.2849e-4 )</td>
</tr>
<tr>
<td>( \Delta t = \Delta x = 0.004 )</td>
<td>( L_{r_e} = 1.3769e-5 )</td>
<td>( L_{r_e} = 9.9817e-5 )</td>
<td>( L_{r_e} = 2.9006e-05 )</td>
<td>( L_{r_e} = 1.2849e-4 )</td>
<td>( L_{r_e} = 2.9344e-5 )</td>
</tr>
<tr>
<td>( \Delta t = \Delta x = 0.002 )</td>
<td>( L_{r_e} = 3.0017e-6 )</td>
<td>( L_{r_e} = 2.4877e-5 )</td>
<td>( L_{r_e} = 7.2472e-6 )</td>
<td>( L_{r_e} = 2.9344e-5 )</td>
<td>( L_{r_e} = 2.9344e-5 )</td>
</tr>
</tbody>
</table>
Example 3. We know from [30] that the exact solution of equation (1) is

$$U(x,t) = a_0 \exp (-kx + \int k^2 + \beta \ dt)$$

(8)

where $\delta = 1$, $a_0$, $k$, and $\beta$ are arbitrary, and $\alpha = \frac{\beta}{k}$. Set $a_0 = 2$. Consider equation (1) for $x \in [0,1]$, with initial and boundary conditions as in (8). Table 3 shows the maximum relative errors at $t = 0.5$ and $t = 1$ with $\beta = 6t^5$ and various values of $k$, $\Delta t$, and $\Delta x$. In Fig. 4 and Fig. 5, the function $U$ introduced in relation (8) is plotted with $a_0 = 2$, $\beta = 6t^5$, and $k = 2, 3$, respectively.

Fig. 3. Function $U$ with $k = 4$ in Example 2

Fig. 4. Function $U$ with $a_0 = 2$, $\beta = 6t^5$ and $k = 2$ in Example 3

Fig. 1 shows that the changes of the function $U$ are not large for $x \in [0,1]$ and $t \in [0,1]$. Figs. 2, 3 show that the changes of the function $U$ are large for $x \in [0,5]$ and $t \in [0,1]$. Figs. 4 and 5 show that the changes of the function $U$ are large for $x \in [0,1]$ and $t \in [0,1]$. The numerical errors in Tables 1, 2, and 3 confirm that the proposed method is accurate enough. As far as the author is aware, many researchers have not tested their
proposed methods on the Burger–Fisher equation that the changes of its solution function are large. The Crank-Nicholson method with the JFNK method seems to be a powerful tool for approximating the Burger–Fisher equation that the changes of its solution function are large.

![Graph of U(x,t) with a_0 = 2, \beta = 6t^3 and k = 3 in Example 3](image)

**Fig. 5. Function U with a_0 = 2, \beta = 6t^3 and k = 3 in Example 3**

### 4 Conclusion

In this paper, an implicit finite difference scheme based on the Crank–Nicolson method was used to discretize the Burger–Fisher equation with initial and boundary conditions. The JFNK method was applied for solving the system of nonlinear equations. Despite the big changes in the solution functions, numerical tests confirmed the accuracy of the proposed numerical method.

### Competing Interests

Author has declared that no competing interests exist.

### References


© 2022 Hajinezhad; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
https://www.sdiarticle5.com/review-history/85584